SJCE_HackOverflow – Solution Design & Architecture

Problem Statement:

To build a generic model to collect information about number of people entering a specific place and to predict the number of people for short term(In terms of days) and long term(In terms of months) future.

Solution:

We have designed a 3 step process as our solution.

Step 1 includes collecting the data from hardware components. Next step is to prepare the dataset for forecasting based on daily and monthly frequency. Finally, we predict the values for the corresponding datasets.

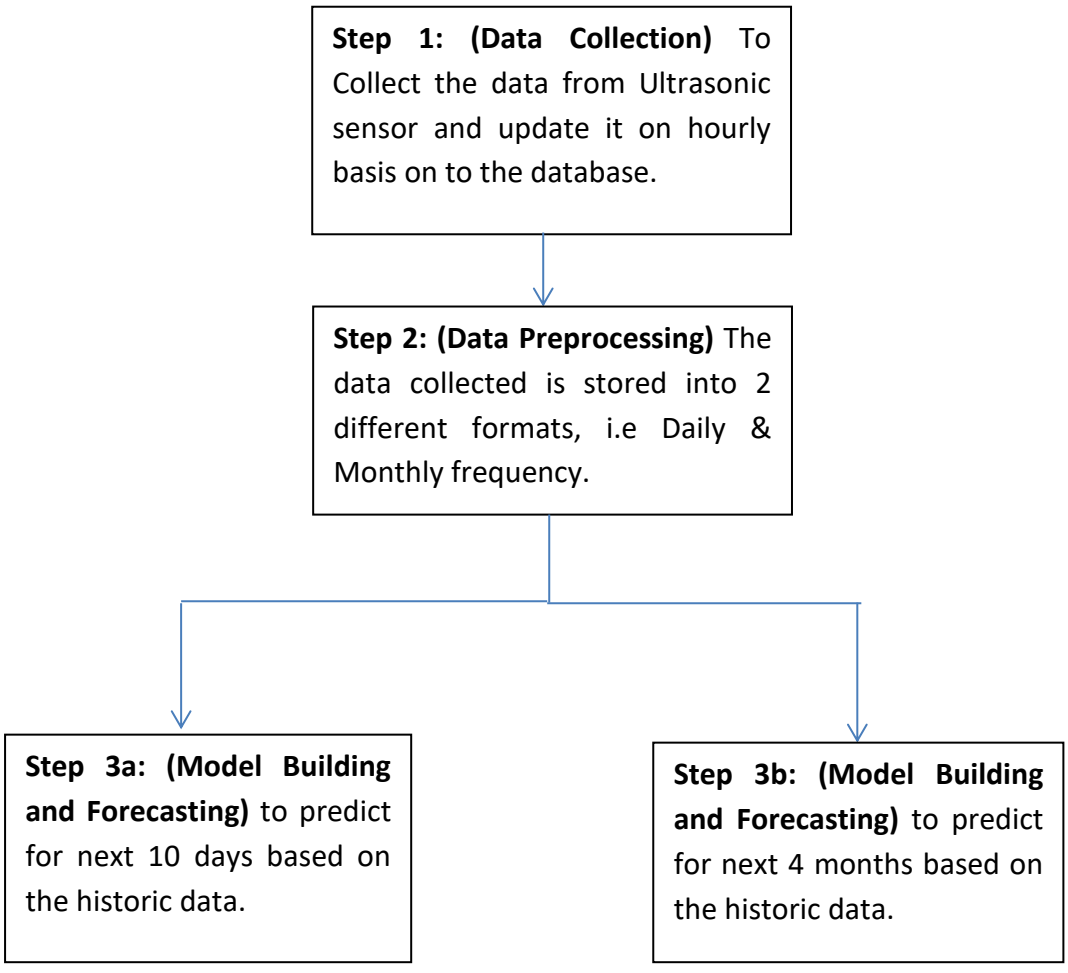

Fig: Block Diagram

Step 1: To Collect the data from Ultrasonic sensor and update it on hourly basis on to the database.

In our project we have used ultrasonic sensor to count the number of people and NodeMCU to temporarily store the count for an hour and then to upload the same count value to the server. Here we have XAMPP server, which creates a server in our PC, and the database is in that server which is created locally.

Ultrasonic sensors emit short, high-frequency sound pulses at regular intervals. These propagate in the air at the velocity of sound. If they strike an object, then they are reflected back as echo signals to the sensor, which itself computes the distance to the target based on the time-span between emitting the signal and receiving the echo.

Working procedure: (Filename – DataCollection.ino)

1. Initialize the pins, variables and assign SSID and password of the Wi-Fi.
2. The NodeMCU gets connected to the Wi-Fi
3. Start timer.
4. Wait for pulse received to be less than threshold value, and then raise a flag.
5. A delay of one second is given so as to give time for crossing.
6. When flag is raised and the received echo signal time runs out of threshold, counter is incremented and flag is reset.
7. The above steps, (4 to 6) are repeated for 1 hour and the count value is sent to the server.
8. Then it is reset.

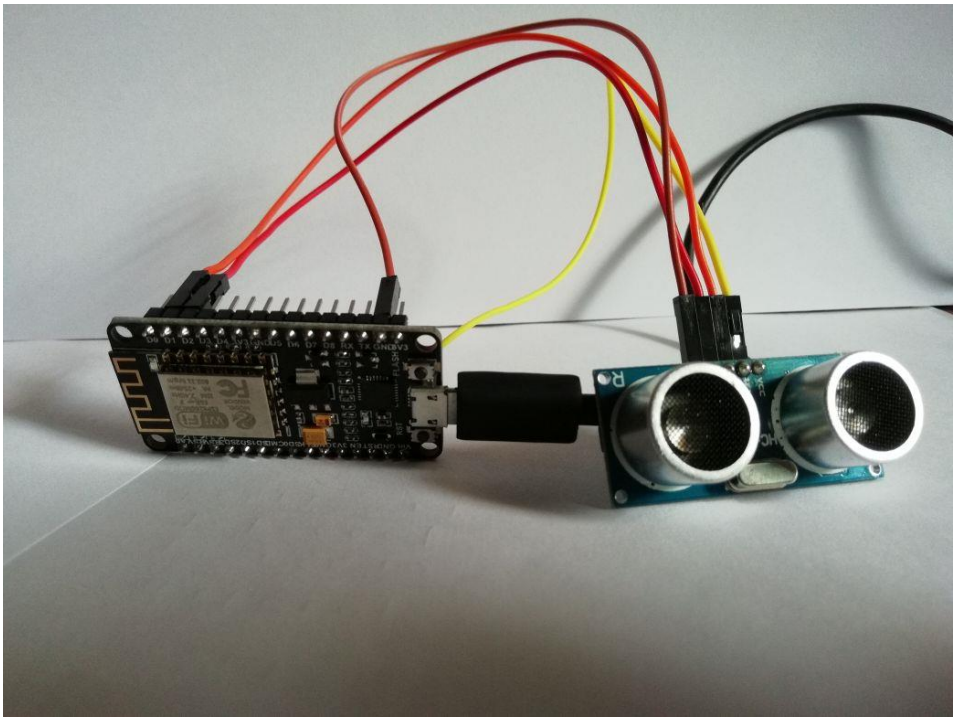

Fig: Hardware designed for data collection.

We believe Machine Learning algorithms are deployable only if we can collect data. So, we have built the hardware for collecting data from real-world.

To upload the count value: (Filename- Writer.php)

- ➔ We have written a Php file, which is in the server. Whenever that Php file is called through the URL with the parameters, it updates the value in the database.
- ➔ Here the NodeMCU triggers that file with the URL and then the count value is passed in the

URL -http://192.168.137.1/writer.php?table=bask1&count=<from NodeMCU>

- ➔ This triggers the Php file and it updates the value of count with timestamp in the database. This can be viewed by PhpMyAdmin application that is preinstalled in XAMPP.

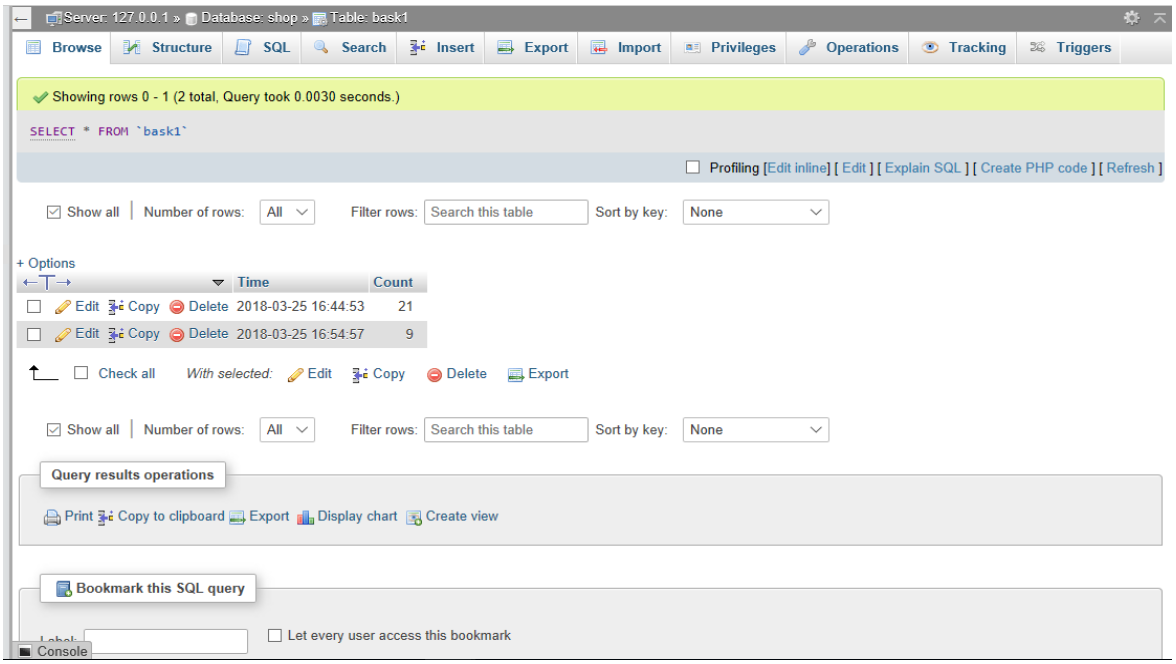

Fig: Screenshot of the Table in the Database

Note: In the above picture, we have shown the sample of collecting the data from hardware at every 10 minutes. When the model is implemented, we can set it 60 minutes and update the database for every one hour.

Step 2: The data collected is processed into 2 different formats, i.e Daily & Monthly frequency.

In this step, the data stored in the database is read and converted into suitable format for the time series model.

(Filenames – DailyDataPreProcess.py, MonthlyDataPreprocess.py)

- ➔ We will re-sample the hourly data into daily and monthly data by summing up the count value.
- ➔ This step has to be done to update the dataset. By updating the dataset we feed in new values as previous data for our model.

Note:

1) Since the time given to us was limited & as we require huge amount of data for forecasting, we couldn't create the dataset with our hardware designed.

2) For illustration of our idea, we have used publicly available datasets.

i) For Daily forecasting – **(Filename-daily_data.csv)**
https://datahack.analyticsvidhya.com/contest/practice-problem-time-series-2/

ii) For Monthly forecasting – **(Filename-monthly_data.csv)**
https://www.kaggle.com/rakannimer/air-passengers/data

Step-3: To forecast new values based on the historic data.

Note: Target Value Count is Continuous variable.

Step 1) Data Visualization (Filename-DataVisualize.py) –

i) Timeplot (Line plot):

A timeplot (sometimes called a time series graph) displays **values against time**. They are similar to x-y graphs, but while an x-y graph can plot a variety of "x" variables (for example, height, weight, age), timeplots can *only* display time on the x-axis.

Timeplots are good for showing how data changes over time. We can use Time Series Plot to look for patterns in data over time, such as trends or seasonal patterns. A time series plot can be used to choose a time series analysis model.

ii)Scatterplot/Hexbin plot:

Hexbin plots can be a useful alternative to scatter plots if the data are too dense to plot each point individually.

Step 2) Choosing a model:

Our goal is to apply time series forecasting for any place. So as to achieve our goal, we require different models for different situations. Hence, we are providing a set of 5 different models for Time Series forecasting, which can be chosen based on the data collected.

1	**Moving Average**	**sma.py**
2	**Simple Exponential Smoothing**	**ses.py**
3	**Holt's Winter Method**	**holt_winter.py**
4	**ARIMA with Grid Search**	**arima.py**
5	**Seasonal ARIMA**	**sarima.py**

Instructions for choosing a model:

1) Observe the data from the Timeplot.
2) Learn about the trend & seasonality of the data from the plot
3) Create a Validation set to choose a model.
4) If the data does not has any trend - Choose Moving Average & Simple Exponential Smoothing.
5) If the data has some trend - Choose Holt's Winter Method(Triple Exponential Smoothing) & ARIMA/ SARIMA (Seasonal- ARIMA)
6) Compare the Results obtained (Preferably RMSE).

We will use the model with least RMSE to forecast new values.

Note: In the next page, we have provided instructions for implementing this solution to Real- world as an Application/Website.

Steps to implement our solution to real world:

1) Deploy the Hardware in the entrance of the place where the Count has to be forecasted.
2) Collect the data and store it in a Database for further usage.
3) Re-sample the data as per the requirement.
4) When the Application/Website is being used for the first time (after collecting the data), an option should be given to automatically calculate RMSE for all 5 models by creating a Validation set. Now, the one with the least RMSE should be selected for forecasting.
5) The model chosen can be used for a specific period of time to forecast new values. (After a certain threshold, step 4 must be repeated to ensure the data still follows the same trend and that particular model gives us the best results)

Note: LSTMs can be used to predict better results, but it would consume lot of resources and it is not feasible to use LSTMs in a standalone machine.

Thank you,
Team Name – SJCE_HackOverflow

Team members –

1) Amogha Subramanya D A
2) Anurag A S
3) Balachandra H N